www.cadanda.com Computer-Aided Design AND APPLICATIONS™

Skeleton Based Modeling based on TGM

Katsutoshi Ootsubo[1], Kazuya G. Kobayashi[2] and Hidehiko Yamamoto[3]

[1]Gifu University, otsubo@gifu-u.ac.jp
[2]Toyama Prefectural University, kobayasi@pu-toyama.ac.jp
[3]Gifu University, yam-h@gifu-u.ac.jp

ABSTRACT

Skeleton based modeling is popular as a deformation method for CG character animation, and have many variations. But the calculation cost and the intuition of operation are trade-off at any conventional method. In this paper, we propose a new skeleton based modeling for 3D mesh models. In this method, user can deform an arbitrary mesh like a rubber along with the skeleton and get the result in interactive speed. This method is composed of 3-step algorithm. In the 1st step, a user makes skeletons on the target mesh. In the 2nd step, a user defines a correspondence between mesh and skeleton. In the 3rd step, a user deforms the skeletons, and the deformed mesh is outputted automatically. In this method, we use extended TGM as deformation structure of skeleton. This method achieves 3D mesh deformation for CG character animation and conceptual design of industrial product.

Keywords: skeleton based modeling, character animation, tensegric modeling.
DOI: 10.3722/cadaps.2012.61-69

1 INTRODUCTION

Skeleton based modeling is an intuitive shape deformation method, and various kinds of it are used in the field of 3D-CG in recent years. “Skeleton” is a low dimensional shape of the modeling target, and the target shape is deformed by stretching or bending it.

The deformation algorithms of conventional approaches are categorized as physical simulation or geometrical procedure. The former is typically based on spring-mass system or FEM [1][2], and achieves the intuitive deformation based on stiffness of ideal materials. However, the computation is basically intensive and unstable for the large deformation. In addition, the proper parameters for a desired result are nontrivial, and the establishment of relationship between a target shape and the skeleton is given automatically according to the implicit rules. The latter is based on geometric procedure; FFD [3][4], t-FFD [5], sketch-based deformation [6] and mean value coordinates [7]. In these methods, the deformation is achieved by the stable computation at the expense of intuitiveness like stiffness, and the relationship is easy to understand to a novice user. So the modeling operators based on them are implemented in some commercial software for shape modeling.

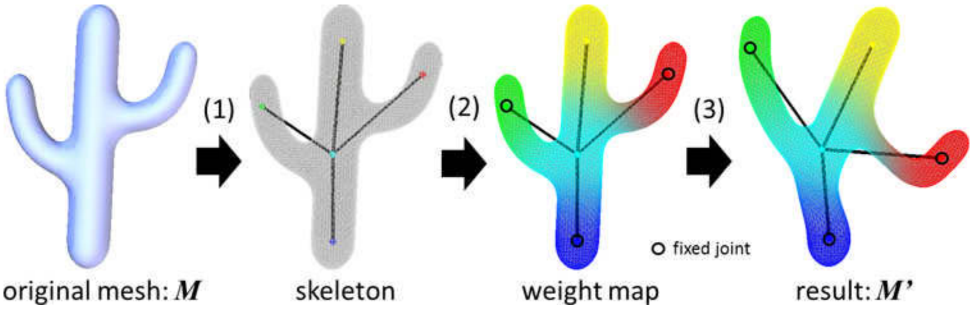

Fig. 1: Deformation steps: generation of a skeleton from a mesh (1), establishment of the relationship (2), and modification of the skeleton and deformation of the mesh (3).

Our proposal method is a new skeleton based modeling based on TGM [8], and the deformation target is represented as arbitrary triangular mesh. Here, Fig.1 shows the deformation steps. When a user wishes to deform an original mesh *M* into *M'*, following 3-steps are executed shown by the black arrows in Fig.1.

(1) A user designs an initial skeleton by specifying arbitrary 2D position on the system window. System generates the TGM related the skeleton automatically.

(2) A user specifies skeleton's effect of deformation against the mesh. The system makes relationship between the skeleton and the mesh automatically according to the effect distribution in a 3D field.

(3) A user modifies the skeleton. The system deforms the mesh *M* into *M'* automatically according to the deformed skeleton and the relationship. A user can repeat the above steps until *M'* reaches the desired shape.

In our system, all operations are achieved by mouse inputs on the system window. The system does not require a novice user of any cumbersome operations like settings of calculation parameter. TGM works as a deformation structure of the skeleton, and represents pseudo stiffness. The mesh is associated to the skeleton via the joints, and the deformed shape is determined uniquely by linear mapping. The calculation is stable, and the result can be seen interactively according to the operations. So, our method has no trade-off between the computational cost and the intuitiveness of the deformation.

In our research, we have developed the prototype system implemented the proposal method. In this paper, we describe the algorithms of our method, and declare the effectiveness by showing the examples given by the system.

2 STGM: SKELETON TGM

TGM is a pseudo physical model based a kind of architectural structure "tensegrity" [9]. TGM is composed of the elements load only pure tension (tensile parts) and pure compression (prestress parts), and achieves self-stabilization by the external forces and the geometrical constraints. TGM has high geometrical stability instead of physical validity like FEM. Tensegric modeling is a pseudo physical deformation method by using TGM as deformation structure of triangular mesh, and achieves rubber-like deformation of the mesh stably in interactive speed. In this method, a unit structure assigned with a face, and the geometry (vertex coordinate) is controlled by the unit. Unit have homeomorphism against the mesh, the adjacent units share the element parts each other. In contrast, because skeleton based modeling needs 1-dimentional deformation structure, conventional TGM can't be applicable for it.

Therefore, we extended TGM unit for skeleton based modeling, and named it "Skeleton TGM (STGM)". STGM has a 1-dimentional pseudo connectivity of unit and gives skeleton pseudo stiffness. Here, the element parts and the topologies are shown in Fig.2. A unit of STGM is composed 6-prestress parts and 27-tensile parts, and the former parts are connected each other via the latter parts. Adjacent units are geometrically constrained at the "connection loop". A unit has 2-connection loops,

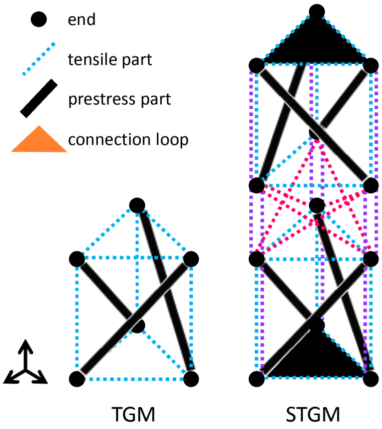

Fig. 2: Unit of conventional TGM and STGM

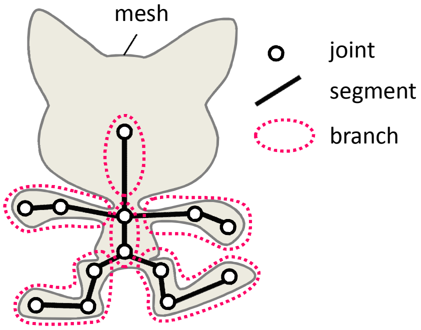

Fig. 3: Entities of a skeleton

and the loop is composed of 3-tensile parts (=ends). In addition, unit and the continuous structure have a feature similar to the conventional TGM about self-stabilization.

3 PROPOSED METHOD

Here, we overview the proposal method, and describes the details of the algorithms.

3.1 Overview

In skeleton based modeling, establishment of relationship between skeleton and the target shape is the most difficult problem to achieve desired deformation. In our method, generation of a skeleton and the establishment of the relationship are calculated based on the drawing information of the target shape and some user inputs. The inputs are given by the operation of some pointing devices like mouse, and the results are reflected in the visual interactively. A user can get a desired relationship intuitively in few steps. In the system, user inputs are initial skeleton, the relationship, and modifying information of the skeleton, and the system outputs the deformed mesh.

3.2 Generation of Skeleton

In this method, “skeleton” is represented by line segments, and is composed of “joints” and “segments” (Fig.3). “Branch” indicates a loop of segments, and has no limits in the topology and the open state.

User specifies positions of joints, and the system generates segments connects them each other. The system uses OpenGL [10] as graphics library. A position of a joint is intermediate of a local shape of the target, and is calculated by using “selection” which is one of the functions. The branch is given automatically by the system at the time segment generates a junction, and is handled as group entity to deform the skeleton.

After that, the system generates STGM along the skeleton. A unit is assigned to a segment as shown in the Fig.4, and the ends are located on the circle centering on a joint. Ends of a connection loop and the joint are geometrically-constrained each other. If a joint is free, the position is given by the simulated ends. Otherwise, the positions of the ends are given by the fixed joint. Here, $r = c \cdot ls$ is radius of a circle, c is a coefficient (usually 1.0) controls pseudo stiffness of a unit, and ls is average length of all segments in a skeleton. In addition, adjacent units are geometrically-constrained each other via the connection loops to keep the same gravity center and the angles between the loops. So STGM simulates a pseudo 1-dimensional deformation structure of the skeleton.

In the deforming phase, a user modifies the skeleton by translating or rotating the joint. The system simulates the behavior of STGM based on the modified skeleton, and calculates the deformed skeleton from the results.

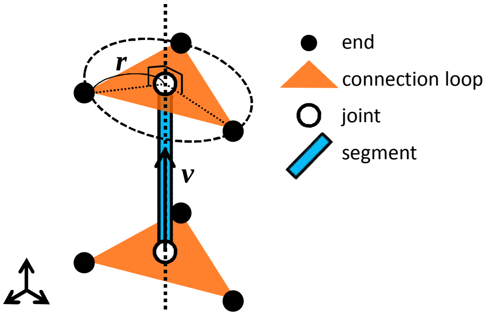

Fig. 4: Geometrical relationship between a unit and a segment.

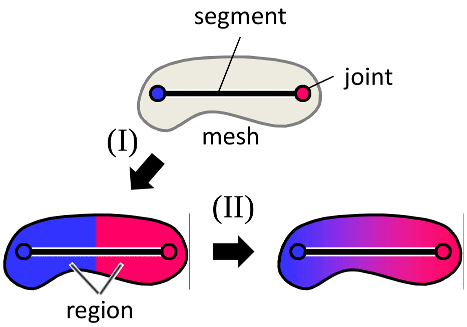

Fig. 5: Weighting: region segmentation (I) and smoothing (II).

3.3 Weighting

In our method, establishment of relationship between a skeleton and a mesh is composed of "parametrization" and "weighing" (Fig.5).

"Parametrization" is calculation process of the local coordinates of the vertices, and the system is defined by the related skeleton. "Weighting" is calculation process of deformation weights by segments at vertices, and is composed of the process; (I) region segmentation and (II) smoothing. In Fig.5, weights are shown by gradation on the mesh.

In the deforming phase, coordinates of the vertices are restored based on the parameters, and deformed mesh is given by blending the coordinates according to the weights.

3.3.1 Region segmentation

Region is topologically-continuous vertices correspond to each joint, and weights of the inner vertices are set to 1.0 (100%). The region is extracted according to the potential values at the vertices by topological searching on the mesh. Fig.6 shows an algorithm of region segmentation, and it is composed of the following 4-steps.

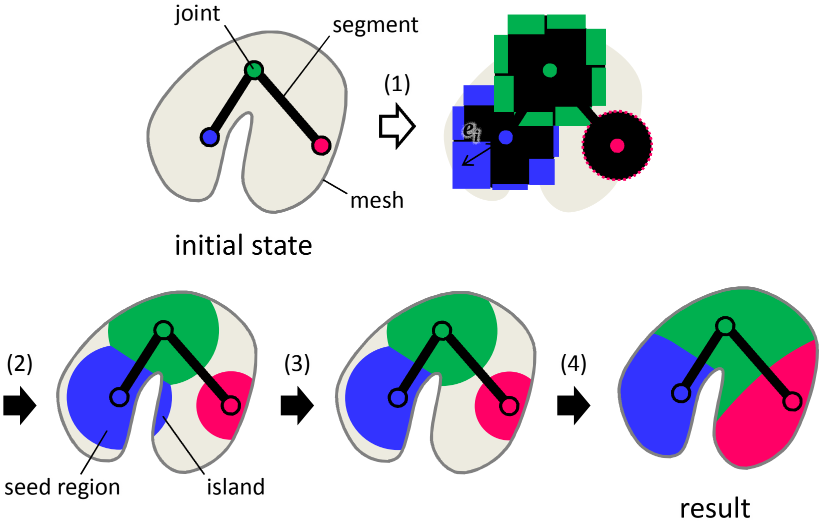

Fig. 6: Region segmentation: definition of effect range (1), extraction of seed regions (2), exclusion of the islands (3), and growing of the seed regions (4).

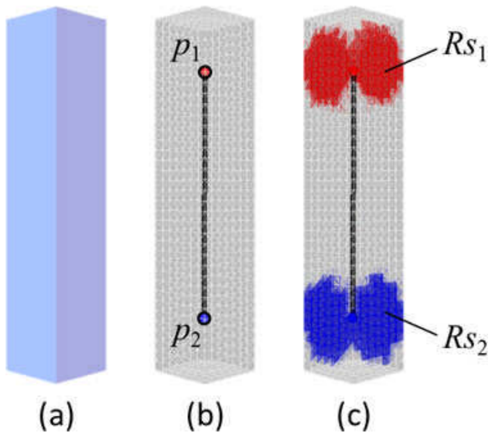

Fig. 7: An example of the seed region: mesh (a), joints of a skeleton (b) and the seed regions (c).

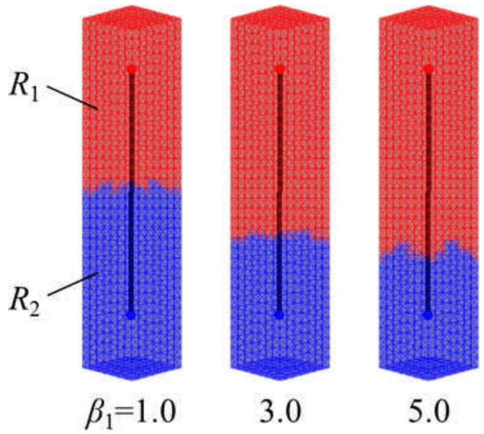

Fig. 8: The final regions (β_2=1.0).

(1) When a user specifies joints, the system executes face-based selection. Then the nearest 2-faces ("seed faces") are found, and the system recorded the faces and the vertices ("seeds").

(2) The system extracts "seed regions" Rs_i (i=1,2,…,l) according to the distribution of "seed potential". Seed region is a joint-dependent region with the max value (>0) at the vertex, and is starting region of the segmentation. The regions are extracted by topological searching starts from the seeds.

Seed potential is the index decides ranges of seed regions, and the parameter is defined by distance from joint to vertex. E_i is a seed potential at a vertex q, and is defined as Eqn. (3.1).

$$E_i = 1 - \frac{d_i}{e_i}, \quad e_i = \alpha_i \cdot r_i \tag{3.1}$$

Here, d_i is the Euclidean distance from a joint p_i to q, r_i is the maximum distance from p_i to the seed faces and the geodesic line between them calculated by dijkstra's algorithm [11]. α_i is user-defined coefficient (usually 1.0), and e_i means threshold of $E_i > 0$.

(3) Sometimes multiple regions against a joint might be found in the previous step. We think the regions except one region are unwanted for a user. We call them "islands", and the system excludes all of them in this step. The index value is $X = \Sigma_{ii} a_{ii} E_{ii}$ (ii=0,1,…,number of inner vertices in a region Rm). Here, a_{ii} is total area of the adjacent faces of an inner vertex q_{ii}, and E_{ii} is total seed potential of Rm. The islands are the region except the area has the maximum value.

(4) The system grows the seed regions to outside according to the distribution of "growth potential", and extracts the final region R_i. When all vertices are included in any regions, the region segmentation is over. Growth potential is the index decides ranges of final regions, and the parameter is defined by geodesic distance from a seed region to a vertex. A growth potential G_i at a vertex q is defined as Eqn. (3.2).

$$G_i = \frac{g_i}{g_i + t_i}, \quad g_i = \beta_i \cdot ls \tag{3.2}$$

Here, t_i is the geodesic distance from Rs_i to q, ls is average length of the all segments, β_i is user-defined coefficient, and g_i means the growth power of Rs_i.

Here, Fig.7 shows an example of the seed region, and Fig.8 shows the final regions.

3.3.2 Smoothing

Smoothing is a blending process of the weight. In this process, the calculation parameters are adjacent vertices around the target vertex. The vertices neighbor the target topologically within a certain range

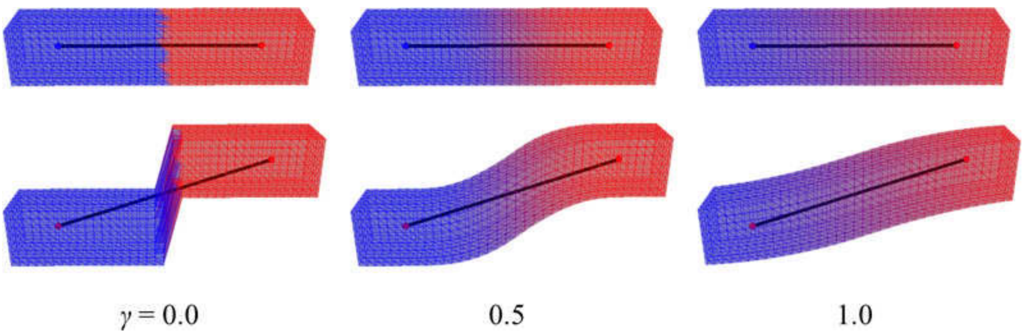

Fig. 9: Weight maps and the deformation results: initial mesh and skeleton (top), and the deformed meshes based on different γ (bottom).

via the edges. Because this process makes the multi-joints contribute to restore of the coordinates of a vertex at boundary of the regions, the deformation is executed smoothly. The blended weight of a vertex q is defined as Eqn. (3.3).

$$wt = \frac{a_j D_j}{\sum_j a_j D_j}, \quad D_j = 1 - \frac{t_j}{\Gamma}, \quad \Gamma = \gamma \cdot ls \tag{3.3}$$

Here, a_j (j=1,2,…,m) is total area of the adjacent faces of the adjacent vertex q_j, t_j is geodesic distance from q to q_j, and γ (≥ 0) is user-defined coefficient. Γ is a searching range of the adjacent vertices, and means the threshold of the blending. After the deformation, the restored coordinates q' is calculated by weighted average of the restored coordinates given by the related joints.

Fig.9 shows an example of weight maps and the deformations. In this example, gradations on the mesh indicate the weight distributions. As shown in this figure, the smoothness of the deformed shape improves as γ increases.

3.4 Parametrization

In the parametrization, the local coordinates (u_k,v_k,w_k) (k=1,2,…,n) of a vertex are calculated from the world coordinates $q(x,y,z)$. The system Σ_k (U_k,V_k,W_k) depends on the related joint p_k, and is a Cartesian coordinate system whose origin is located on p_k. q is represented by affine mapping of (u_k,v_k,w_k), and is defined as Eqn. (3.4)-(3.7). Fig.10 shows a geometrical relationship between q and Σ_k.

$$q = p_k + u_k U_k + v_k V_k + w_k W_k \tag{3.4}$$

$$U_k = \frac{p_k - e_{k,1}}{\left| p_k - e_{k,1} \right|} \tag{3.5}$$

$$V_k = \frac{W_k \times U_k}{\left| W_k \times U_k \right|} \tag{3.6}$$

$$W_k = \frac{U_k \times (e_{k,2} - p_k)}{\left| U_k \times (e_{k,2} - p_k) \right|} \tag{3.7}$$

After the deformation, the restored world coordinates by Σ_k' : q_k' is given by the reverse operation of the parametrization. The final restored world coordinates q' is calculated by Eqn. (3.8).

$$q' = \frac{\sum_k wt_k q_k'}{\sum_k wt_k} \tag{3.8}$$

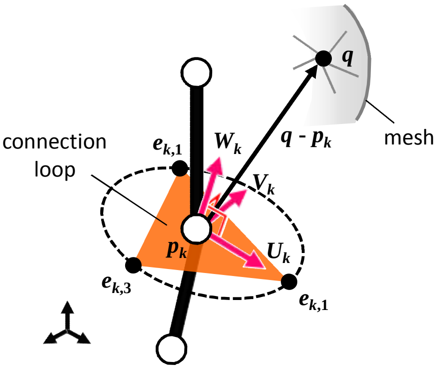

Fig. 10: A point q of a mesh and the local coordinate system $\Sigma_k(U_k,V_k,W_k)$ on the joint p_k.

In the method based on physical simulation, shape determination of the elements is intensive calculation. The total cost is mainly controlled by the elements number, the cost of the reverse operation is negligible compared with it. So the total cost of our method can be compare with the conventional method by the element numbers. In the conventional method, the elements number is controlled by the number of the faces. In contrast, it is controlled by the resolution of the user-defined skeleton in our method. Moreover, our method is differ from control line deformation [15], because Σ_k depends on not a segment but a joint whose geometry is controlled by neighboring STGM units.

4 EXAMPLES

In our research, we developed a prototype system implemented the proposal method, and verified the effectiveness for our objective by the computational experiments. The development and the execution environments are shown as follows.

Language: C#, Additional library: OpenTK v.0.9.1 [12], .NET Framework v.2.0 [13], CPU: Intel Core i3 540, Memory: 10.0GB, Graphic: NVIDIA Quadro FX 580, OS: Windows 7 Ultimate 64Bit

Barr represented primitive operations for intuitive shape modeling; "bending", "twisting" and "pulling" [14]. Fig.11-13 demonstrates the examples given by the system, and the deformations correspond to the operations are applied. These figures show the skeleton and the weight (top), the mesh (bottom), the initial states (left) and the results (right). In the captions of these figures, "FPS" is frame per second, and means an update average of calculation result during the executions. In a 3D shape modeling, we think a user can feel an operation interactive when the FPS is over 10.

Fig.11 shows a "bending" example. In this figure, 3-joints represented as circle points are fixed, and those are translated. As shown in the figure, the relationship between the skeleton and the mesh is given along the segments locally, and changes smoothly at the boundaries. The segments are deformed greatly and locally, and deformed globally to keep the initial relative geometry (the position and the posture) as much as possible. The mesh is deformed smoothly along them, and the result shows "bending" can be represented by the system.

Fig.12 shows a "twisting" example. In this figure, 2-joints at the end of the skeleton are fixed, and the one is rotated around the vertical direction of the figure. As shown in this figure, the mesh is twisted smoothly to interpolate the transformation keeping the ridge lines. However, the deformed shape seems to be non-uniform between the both ends, and we suspect the disconnection between the components of the adjacent STGM-units as the cause of it.

Fig.13 shows a "pulling" example. In this figure, 3-joints are fixed, and the 2-joints of them are translated. As shown in this figure, the segments connect between the fixed joints changes the lengths and the postures, and the forepaws and the body of the mesh ("tiger") are pulled downward smoothly. In addition, the head turns up according to change of the adjacent segments, and it seems to be intuitively plausible result.

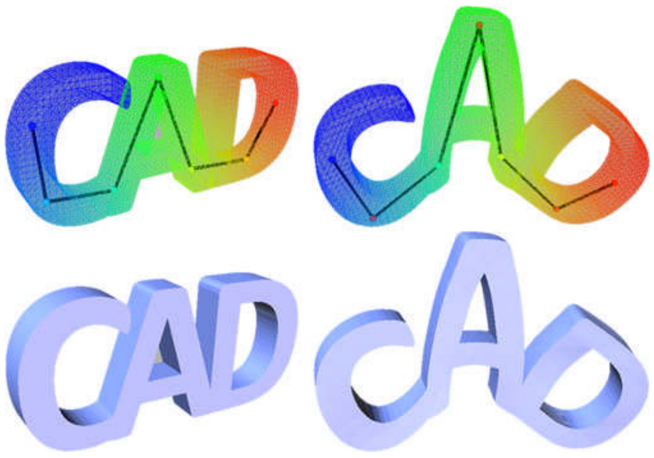

Fig. 11: Bending (faces=7762, segments=6, β_i=1.0 (i=1,2,…,6), γ=1.5, FPS=11.9).

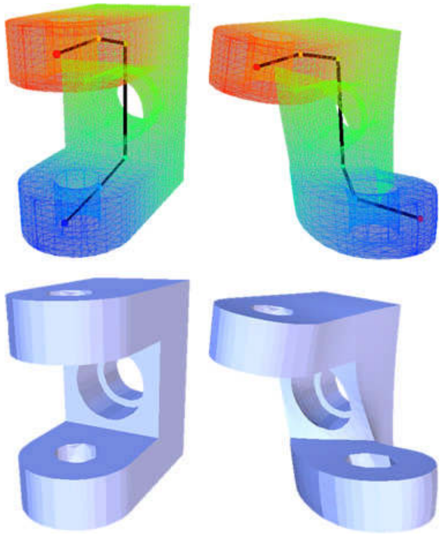

Fig. 12: twisting (faces=6038, segments=5, β_i=1.0 (i=1,2,…,5), γ=1.5, FPS=12.9).

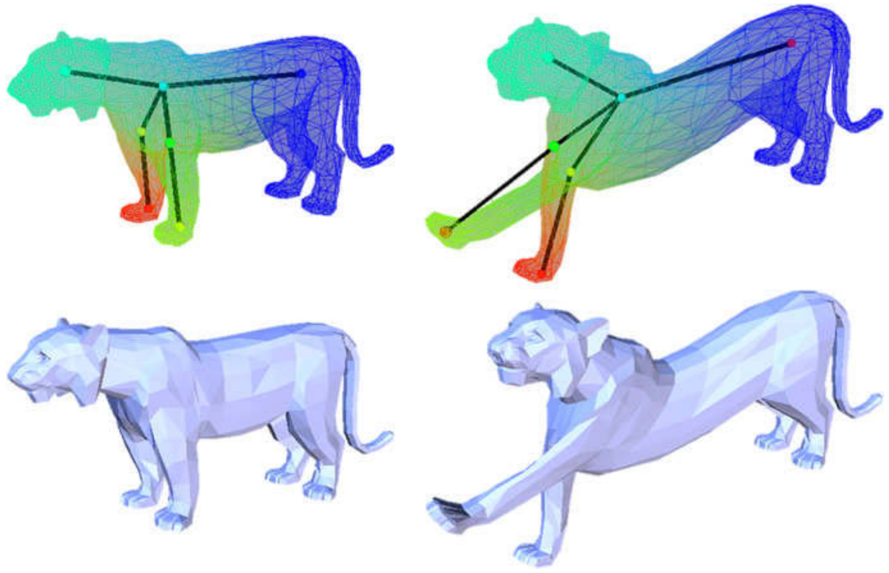

Fig. 13: Pulling (faces=1908, segments=6, β_i=1.0 (i=1,2,…,6), γ=1.0, FPS=47.9).

In all examples, the deformed shapes were updated in interactive speed, and the final shapes were given by few step operations. The modeling shapes are deformed smoothly like rubber, and the calculations have no vibration or divergence through the operations.

5 CONCLUSIONS

In this paper, we proposed a new skeleton based modeling based on 3D-mesh and TGM, and verify the effectiveness through the experiments by using the prototype system.

The relationship between the mesh and the skeleton is established intuitively according to the mesh topology by the user-defined joints. The skeleton behaves as if it has a certain kind of pseudo stiffness, and the local deformation affects the global shape. The result mesh is deformed smoothly along with the skeleton like rubber.

The mesh geometry is updated with interactive speed in the above examples. The results are given by few number of deformation elements because of the linear mapping. In addition, STGM simulation and the mapping is stable calculation, so the large deformation is achieved stably by this method.

In the future, possible improvements and desirable extensions about this method are shown as follows.

- Resolution of the problem in twisting operation;

  Units of STGM are geometrically constrained between the neighbors. We think the rotation of the unit is not propagated enough over the connection loop by the constraint. So that can be solved by developing the interconnect structure of STGM.

- Improvement of the calculation speed;

  In this method, because we use topological search as basic algorithms in the pre-process of the operation (weighting), the calculation cost against dense mesh is not negligible. The appropriate selection of the target or more efficient search method can reduce the cost.

- Application for 2D;

  The proposal method is applicable for 2D by extending STGM or mapping to 2D shape (raster or vector image). We think the 2D method is effective for CG animation.

REFERENCES

[1] Capell, S.; Green, S.; Curless, B.; Duchamp, T.; Popovic, Z.: Interactive skeleton-driven dynamic deformation, Proc. of the ACM SIGGRAPH '02, 2002, 586-593. DOI:10.1145/566570.566622

[2] Shi, X.; Zhou, K.; Tong, Y.; Desbrun, M.; Bao, H.; Guo, B.: Example-based dynamic skinning in real time, ACM Transactions of Graphics, 27(3), 2008, Article 29. DOI:10.1145/1360612.1360628

[3] Forstmann, S.; Ohya, J.; Grimberghe, A., K.; McDougall, R.: Deformation styles for spline-based skeletal animation, Proc. of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2007, 141-150.

[4] Yoshizawa, S.; Velyaev, A. G.; Seidel, H. -P.: Free form skeleton-driven mesh deformations, Proc. of the 8th ACM symposium on Solid modeling and applications, 2003, 247-253. DOI:10.1145/781606.781643

[5] Kobayashi, K., G.; Ootsubo, K.: Shape Modeling with t-FFD and Tensegric Modeling, JICAST 8th Joint International Conference, 2004, 125-131.

[6] Koh, Y.; Garland, M.: Sketching mesh deformations, ACM Transactions on Graphics, 24(3), 2005, 147-154. DOI:10.1145/1073204.1073291

[7] Ju, T.; Zhou, Q. -Y.; Panne, M.; Cohen, -O. D.; Neumann, U.: Reusable skinning template using cage-based deformations, ACM Transactions of Graphics, 27(5), Article 122, 2008. DOI:10.1145/1409060.1409075

[8] Kobayashi, K. G.; Ootsubo, K.: Shape modeling for arbitrary mesh models, Digital engineering workshop – the 5th Japan-Korea CAD/CAM Workshop, 2005, 168-171.

[9] The Buckminster Fuller Institute: http://www.bfi.org/, Buckminster Fuller Institute.

[10] OpenGL FAQ, http://www.opengl.org/resources/faq/technical/selection.htm, Khronos Group.

[11] Dijkstra, E. W.: A note on two problems in connexion with graphs, In numerische mathematic, 1(1), 1959, 269-271. DOI: 10.1007/BF01386390

[12] The Open Toolkit Library | OpenTK, http://www.opentk.com/.

[13] .NET Framework Developer Center: http://msdn.microsoft.com/en-us/netframework/, Microsoft Corporation.

[14] Barr, A., H.: Global and local deformations for solid primitives, Proc. of SIGGRAPH '84, 1984, 21-30. DOI:10.1145/800031.808573

[15] Kanai, T.; Toki, K.; Hashimoto, Y.; Suzuki, H.; Terasawa, M.; Kimura, F.: 3D shape deformation based on control lines and its applications, Symposium of Graphics & CAD + Visual Computing, 1997, 9-16. (in Japanese)